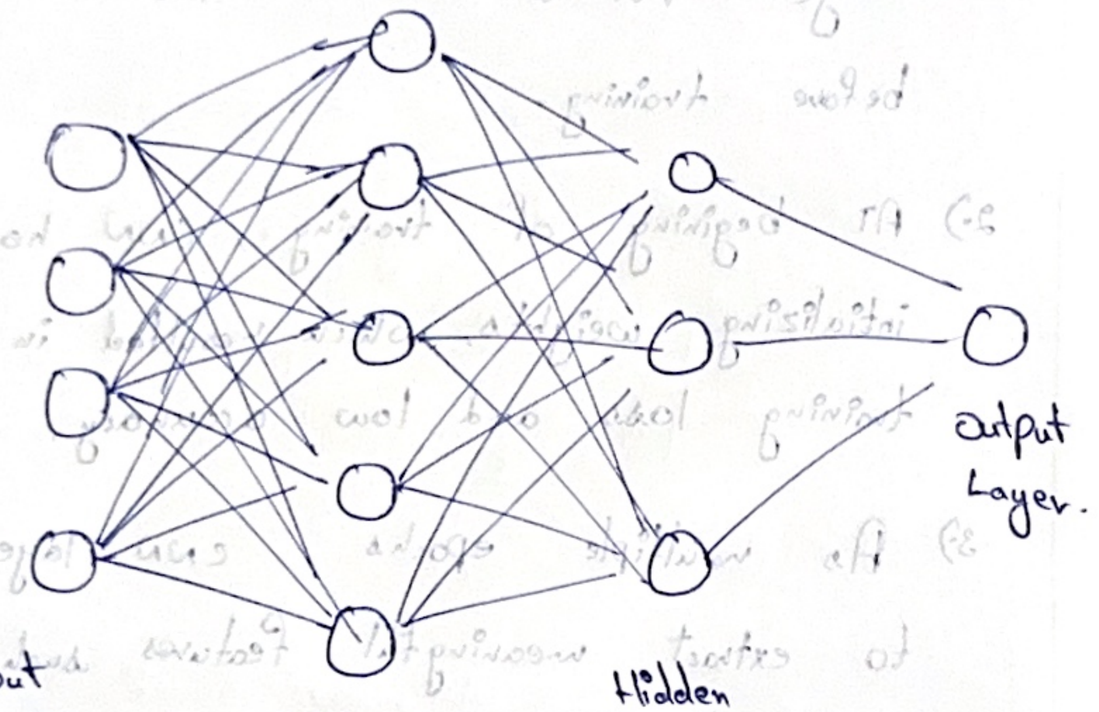
RNN Architecture



Exp :9:-

Build a Recurrent Neural Network.

Aim:-

To build train a recurrent Neural Network (RNN) for Sequence modelling.

Objectives :-

- To understand the working principles of RNN
- To Preprocess Sequential Data for RNN.
- To design and implement an RNN using Pytorch.
- To train the RNN model and evaluate the Performance.
- To analyze the outputs.

Pseudocode :-

1.) Start.
2.) Input necessary libraries.
3.) Load dataset.
4.) Preprocess dataset.
   -> clean data
   -> Taking / encode sequences.
   -> Pad / trangate sequences.
   -> split into training and validation sets.
5.) Define RNN model.
   -> Initialize sequential model.
   -> Add embeddy layer or input layer.
   -> Add simple RNN layers (s).

6.) Compile model.

-> Specify optimizer.
-> Specify loss function.
-> Specify evaluation

7.) Train model.

-> feed training data.
-> Specify batch size and no. of epochs.
-> validate any validation set.

8.) Evaluate model.

-> Test on unseen data.
-> Print accuracy / loss.

9.) End.

Output:-

Accuracy = 0.7174.

marco precision = 0.7197.

marco recall = 0.7174.

marco F1 score = 0.7167.

Pre class precision = [0.74181498 0.69752907]

Pre class Recall = [0.66704 0.76784].

Pre class F1 = [0.70244313 0.73099772].

Confusion matrix :-

[[8338 4162]
[2902 9598]].

classification report :-

	Precision	recall	f1-score	support
Neg	0.74	0.67	0.70	12500
Pos	0.70	0.77	0.73	12500
accuracy				
macro avg	0.72	0.68	0.72	13500
weighted avg	0.72	0.67	0.73	13500